risti
Revista Ibérica de Sistemas e Tecnologias de Informação
Revista Ibérica de Sistemas y Tecnologías de Información

Recebido/Submission: 31/10/2019
Aceitação/Acceptance: 20/02/2020

Plataforma de análisis de datos para la evaluación de desempeño de software

Silvana de Gyves Avila[1], Patricia Ortegon Cano[1], Ayrton Mondragon Mejia[1], Ismael Solis Moreno[1], Arianne Navarro Lepe[1], Gloria Eva Zagal Dominguez[1]

silvana.degyves@ibm.com, patricia.ortegon@ibm.com, ayrton.mondragon1@ibm.com, ismael@mx1.ibm.com, arianne.navarro@ibm.com, gloria.eva.zagal@ibm.com

[1]IBM, México Software Lab, Carretera al Castillo 2200, 45680, el Salto Jalisco, México

DOI: 10.17013/risti.36.50–64

Resumen: El desempeño es un parámetro importante en los procesos de evaluación de software. Es usado como punto de diferenciación entre competidores. El aseguramiento del desempeño no es trivial, ya que requiere la ejecución de pruebas exhaustivas y el análisis de grandes volúmenes de datos. Normalmente, se usan soluciones comerciales para producir métricas de desempeño. Sin embargo, estas son de propósito general y requieren de esfuerzo para ser adaptadas a casos particulares. En este artículo presentamos DDP, una plataforma para analizar datos de pruebas de desempeño. DDP utiliza tecnología de Big Data para recolectar, almacenar, procesar y analizar resultados de desempeño de una manera integrada. Demostramos el uso exitoso de DDP en la evaluación de "Spectrum Scale", una solución de almacenamiento definido por software. Ilustramos el uso DDP en el análisis de pruebas de regresión para verificar y validar la calidad de las nuevas versiones creadas durante el proceso de desarrollo.

Palabras-clave: calidad de software; desempeño; big data; ciencia de datos.

A Data Analysis Platform to Evaluate Performance During Software Development Process

Abstract: Performance is one of the most important parameters to consider during the software development process. It is used as competitive advantage among similar solutions. Assuring expected performance levels is not trivial since it requires to run exhaustive tests and the analysis of Big Data. Normally, companies rely on commercial solutions to produce performance analytics. However, these require a significant effort to be adapted to particular performance use cases. In this paper, we describe DDP, a platform to analyze performance testing data. DDP uses Big Data technology to collect, store, process and analyze performance results in an integrated way. We demonstrated the successful use of DDP evaluating the performance of "Spectrum Scale", a software defined storage solution. We illustrate the use of DDP analyzing data from performance regression tests to verify and validate the quality of new versions build during the development process.

Keywords: software quality; performance; big data; data science.

1. Introducción

El desempeño se considera como una de las dimensiones no-funcionales más importantes en cualquier proceso de desarrollo de software (Kurian, 2018). Mantener o mejorar los niveles de desempeño, como ventaja competitiva, es fundamental para lograr la satisfacción del cliente y los objetivos del mercado. Sin embargo, garantizar que un componente de software ofrezca el desempeño esperado se ha convertido en una tarea compleja y desafiante, junto con la complejidad de los entornos en los que está destinado a ejecutarse (Aleti, 2018). Los sistemas ciber-físicos han evolucionados de servidores stand-alone a entornos distribuidos masivos, compuestos por varios clústeres y cargas de trabajo complejas, que interactúan entre sí. Para que las compañías de software evalúen un subconjunto representativo de escenarios realistas, se requiere de herramientas avanzadas que no solo automaticen la ejecución de las pruebas de desempeño, sino que además manejen grandes volúmenes de datos producidos durante el proceso de evaluación (Kurian, 2018). Los datos de desempeño son complejos y provienen de diversas fuentes, por ejemplo: logs, benchmarks y archivos de configuración, cada uno con distintos niveles de estructura. Por lo tanto, analizar estos datos y producir observaciones, requiere de un conjunto de tecnologías y esfuerzo significativo, para llevar los datos de desempeño a través de su ciclo de vida. Comúnmente, se utilizan paquetes estadísticos como, R-Studio (Dessau, 2008), SPSS (Aldrich, 2018) y Cognos Analytics (Huijgens, 2018). Sin embargo, las etapas que se encuentran entre la recolección, almacenamiento, pre-procesamiento y visualización de datos, se llevan a cabo en su mayoría de forma manual, creando silos donde se desperdician datos, y en muchas ocasiones, se pierden. Esto crea la necesidad de confiar en un marco holístico que pueda ayudar a los analistas de desempeño a simplificar la gestión de datos y la visualización de resultados. En este artículo, se introduce la Plataforma de Desempeño Orientada a Datos (Data Driven Performance) DDP, un enfoque que permite integrar la tecnología necesaria para manejar datos provenientes de las pruebas de desempeño realizadas para soluciones de almacenamiento definido por software. DDP está compuesto por ETLs y motores de procesamiento de datos que trabajan con un repositorio de datos altamente escalable e interfaces de usuario interactivas. DDP ha sido implementado para mejorar el proceso de desarrollo de software de Spectrum Scale, un sistema de gestión de archivos de alto rendimiento que se utiliza ampliamente en múltiples compañías a nivel mundial. DDP utiliza Big Data para generar reportes y dashboards sobre las tendencias de desempeño, y así proporcionar soporte visual en la toma de decisiones durante los procesos de desarrollo y pruebas. Nuestro caso de prueba de éxito muestra la flexibilidad de DDP para manejar una amplia variedad de datos de pruebas de desempeño. DDP complementa exitosamente los procesos de desarrollo y pruebas, para así mejorar la calidad del producto ofrecida a los clientes.

La mayor contribución de este artículo es la descripción de DDP, una plataforma para mejorar los procesos de pruebas de desarrollo. También se presenta un caso de uso donde la aplicabilidad de DDP ha sido demostrada y se discuten los principales analíticos implementados. El resto de este artículo se encuentra organizado de la siguiente forma: la sección 2 presenta el marco conceptual, la sección 3 describe la plataforma de desempeño orientada a datos, la sección 4 describe el caso de uso, la sección 5 presenta los analíticos implementados, y la sección 6 incluye nuestras conclusiones y trabajo futuro.

2. Marco Conceptual

2.1. Dashboard, Estadísticos de Desempeño y Benchmark

A continuación, se presentan los conceptos de dashboard, estadísticos de desempeño y benchmark, los cuales serán utilizados a lo largo de este artículo.

- **Dashboard.** Puede definirse como "una herramienta visual e interactiva que muestra en una sola pantalla la información más importante necesaria para lograr uno o más objetivos individuales y/u organizacionales, permitiendo al usuario identificar, explorar y comunicar áreas que necesiten medidas correctivas" (Yigitbasioglu & Velcu, 2012). En el contexto de desempeño, los dashboards han sido ampliamente utilizados por diversas organizaciones y poco a poco han ido reemplazando a los reportes tradicionales (Velcu-Laitinen & Yigitbasioglu, 2012).
- **Estadísticos de desempeño.** Para identificar y analizar un problema de desempeño, es necesario contar con datos estadísticos, los cuales se obtienen a partir de información histórica. Un estadístico de desempeño se define en base al sistema que está siendo evaluado y las dimensiones de interés. Los estadísticos de desempeño pueden incluir elementos de estadística descriptiva y estadística predictiva.
- **Benchmark.** Es un estándar que se utiliza para evaluar el desempeño relativo de un sistema. Es relativo, ya que los resultados de una prueba generalmente cubren un solo aspecto o dimensión. Existen benchmarks orientados a evaluar velocidad de transferencia, procesamiento, lectura y escritura, etc. (Price, 1989).

2.2. IBM Spectrum Scale

Las características de desempeño de un sistema de software, tales como el tiempo de respuesta, rendimiento de la red, escalabilidad y disponibilidad, son atributos de calidad que se vuelven clave en las aplicaciones distribuidas usadas para realizar el análisis de grandes volúmenes de datos. A fin de cumplir con todos estos requerimientos, la forma en que realizamos el almacenamiento de datos se vuelve un factor crítico (Denaro, 2015).

IBM tiene un portafolio robusto de soluciones de almacenamiento definido por software, y uno de los componentes clave de este portafolio es *Spectrum Scale*, un sistema de archivos de clúster que provee acceso concurrente a un solo sistema de archivos o a un conjunto de ellos, desde múltiples nodos (Coyne, 2018). Esto permite acceso de alto desempeño al conjunto común de datos para mantener una solución escalable o proveer una plataforma de alta disponibilidad. *Spectrum Scale* es usado en muchas de las supercomputadoras científicas más grandes del mundo y en aplicaciones comerciales que requieren acceso de alta velocidad a grandes volúmenes de datos (Quintero, 2015).

Cuando una nueva aplicación paralela es desarrollada, una etapa de calibración y ajuste es requerida para evaluar su comportamiento, con el propósito de mejorar el desempeño. Algunos cuellos de botella en el código paralelo solamente pueden ser descubiertos durante el modo de ejecución. Durante este proceso de evaluación se utilizan las métricas tradicionales de desempeño para sistemas distribuidos que son: latencia, descripción del retardo entre una solicitud y la terminación de una operación; rendimiento de la

red, que denota el número de operaciones que pueden ser terminadas en un periodo de tiempo dado; y escalabilidad, que identifica la dependencia entre el número de recursos de sistemas distribuidos que pueden ser usados por una aplicación distribuida. Otras métricas relevantes son actividades de I/O y tiempo de respuesta (Denaro, 2015).

2.3. Aplicaciones Orientadas a Datos en el Desarrollo de Software

En los últimos años, el análisis de datos ha sido aplicado a diferentes campos, desde la ciencia hasta los negocios, y la ingeniería de software no es la excepción. Como podemos ver en diferentes trabajos (Menzies, 2013; Mockus, 2014), las metodologías orientadas a datos han sido aplicadas a todas las fases en el ciclo de desarrollo de software. Y el rol del científico de datos en un equipo de desarrollo de software comienza a ser más popular (Kim, 2016).

Las compañías de software están impulsando un enfoque incesante en la calidad y las pruebas de software, podemos decir que estas áreas están enfrentando un proceso de disrupción en estos momentos. Las tareas más comunes y repetitivas están siendo reemplazadas por frameworks automatizados de pruebas más frecuentemente, y nuevos frameworks de automatización están siendo creados (Yandrapally, 2014).

Un enfoque orientado a los datos en el campo de las pruebas funciona de la siguiente manera: las entradas para las pruebas son almacenadas en una base de datos, un controlador de pruebas toma múltiples conjuntos de datos de entrada y ejecuta los casos de prueba. Cuando las entradas son agregadas o removidas de la base de datos, las pruebas no son alteradas. Una revisión de diferentes trabajos que siguen este enfoque, principalmente orientado a las pruebas de requerimientos funcionales puede encontrarse en (Anbunathan, 2015).

Cuando estamos verificando los requerimientos funcionales de cualquier aplicación, frecuentemente los problemas se pueden rastrear hacia el código o el diseño. Pero cuando se trata de los requerimientos no funcionales, donde el desempeño juega un rol principal, encontrar el problema puede ser más complejo debido a que se involucran diferentes aspectos fuera de la implementación del software. Por ejemplo, la red y las configuraciones de hardware que requieren la participación de diferentes expertos en dichas áreas.

Durante el ciclo de desarrollo, las variaciones en el desempeño pueden ocurrir en muchas ocasiones, pero las pruebas de desempeño son la herramienta para evitar el impacto que dichas variaciones negativas podrían tener en el siguiente entregable del producto de software (Mostafa, 2017). Los desarrolladores requieren detectar regresiones en el desempeño tan pronto como sea posible, para reducir el impacto negativo y el costo que conlleva arreglar dichos problemas. Entonces, existe una necesidad para integrar este tipo de pruebas al ciclo de desarrollo y no dejarlas hasta el final como se realiza comúnmente.

La evaluación del desempeño es usualmente cara de realizar debido a que requiere ejecuciones frecuentes que pueden llevar horas o incluso días. Además, la cantidad de métricas y datos que se generan frecuentemente son complejos y se producen en grandes volúmenes.

La mayoría del trabajo reciente que podemos encontrar relacionado con la automatización de pruebas de desempeño está enfocado en el análisis de los datos recolectados. En (Shang, 2015), se propone un enfoque basado en modelos para automatizar el análisis de las métricas de desempeño, construyendo modelos de regresión para encontrar retrocesos en el desempeño del software. En (Foo, 2015), se presenta otro trabajo basado en el impacto de los diferentes ambientes de ejecución donde las pruebas de desempeño son realizadas, usando modelos estadísticos. En (Berral, 2017), una propuesta de aprendizaje automático es usada para interpretar benchmarks de datos de desempeño en aplicaciones de Big Data.

También podemos encontrar algunos trabajos que tienen como objetivo otros pasos en el proceso de las pruebas de desempeño. En (Mostafa, 2017), se implementa un método para priorizar los casos de pruebas de desempeño, a fin de permitir más ejecuciones en etapas tempranas del desarrollo. En (Kroß, 2016), se presenta una propuesta de arquitectura de DevOps para pruebas de desempeño. En (Okanović, 2019), se presenta una herramienta para crear reportes de análisis de desempeño basados en diferentes intereses.

Recolectar, analizar, modelar y visualizar datos son pasos dentro del proceso de evaluación de desempeño. Sin embargo, ninguna de las soluciones existentes provee la integración de todas las fases de dicho proceso. Herramientas que permitan el análisis de todos estos datos y los presenten a los diferentes tomadores de decisiones, de una manera sencilla son una necesidad primordial.

Las diferentes propuestas descritas en esta sección se enfocan en uno o dos de los pasos del proceso, mientras los pasos restantes son cubiertos usando algunas herramientas comerciales que no siempre son ad-hoc para el sistema que se está probando, o en otros casos, estos pasos son ejecutados de manera manual. Integrar todas las diferentes soluciones existentes puede ser muy complejo y llevar a la pérdida de datos. Por lo tanto, aún existe una necesidad de herramientas que permitan integrar todas las etapas del proceso de evaluación de desempeño en un solo marco de trabajo. En las siguientes secciones describimos DDP, una plataforma para analizar y explotar datos de desempeño en todas las fases de evaluación.

3. Plataforma DDP

La plataforma DDP proporciona a los usuarios un repositorio centralizado que les permite almacenar, organizar y visualizar resultados de desempeño, hacia y desde una sola fuente. Esto ayuda a resolver el problema que se genera cuando se consumen datos de desempeño (tales como: resultados de benchmarks, datos de cargas de trabajo, perfiles de sistemas, configuración de hardware y logs, entre otros), que tienen diversos orígenes y formatos, y que no se encuentran relacionados de forma explícita.

Los componentes que integran la arquitectura de la plataforma de DDP se muestran en la figura 1 y se describen a continuación:

- **ETL (Extract, Transform and Load).** Módulo que se encarga de la lectura, procesamiento y almacenamiento de información relevante obtenida a partir de diversos archivos que incluyen resultados de pruebas de desempeño. Cada

archivo incluye información que debe ser filtrada y estructurada, antes de ser almacenada en el repositorio. Esto facilita el análisis numérico y textual que realizan los otros módulos de la plataforma. Este módulo se encuentra implementado en Python. Para cada uno de los diferentes benchmarks que se ejecutan durante una evaluación de desempeño, existe un analizador que utiliza expresiones regulares (importadas de la librería re de Python), para extraer toda la información relevante. Los valores obtenidos se procesan y almacenan en las tablas correspondientes definidas utilizando modelos de Cassandra.

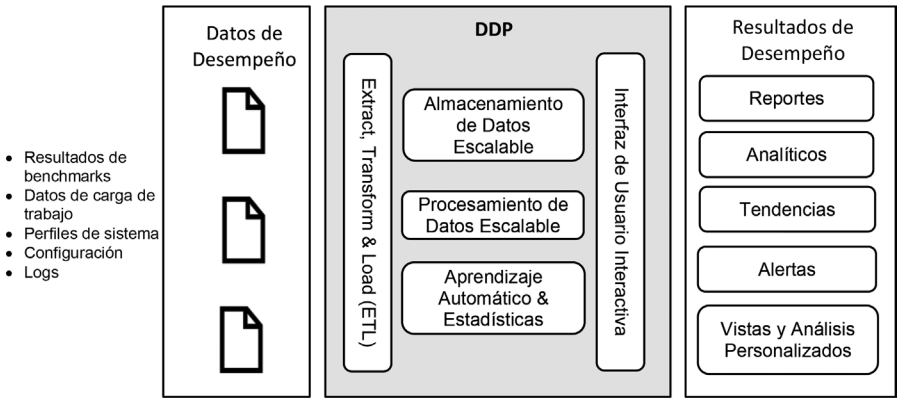


Figura 1 – Arquitectura de la Plataforma DDP

- **Almacenamiento de Datos Escalable.** Repositorio en el cual se almacenan los diferentes datos de desempeño. Este módulo se encuentra formado por dos bases de datos NoSQL: Cassandra (The Apache Software Foundation, 2016) y MongoDB (MongoDB, Inc., 2020). Los datos estructurados se almacenan en las tablas de Cassandra, mientras que los archivos que contienen los datos de desempeño (con el detalle de cada prueba), se almacenan como documentos en MongoDB. La información estructurada se utiliza para la etapa de análisis, mientras que los documentos se utilizan para consulta. En algunos casos, puede ser necesario que un experto humano realice una revisión de detalles contenidos en esos documentos, esto con fines de rastreo de procedencia.
- **Procesamiento de Datos Escalable.** Este módulo procesa los datos de desempeño que se encuentran almacenados en las tablas de Cassandra. El procesamiento de los datos se ejecuta utilizando trabajos de Scala (École Polytechnique Fédérale, 2002-2020) y un clúster de Spark (The Apache Software Foundation, 2018). Para evitar tiempos de respuesta elevados al momento de consultar gráficos y reportes (comparativas de desempeño), todos los resultados obtenidos por el módulo de procesamiento también son almacenados en estructuras de Cassandra.
- **Aprendizaje Automático & Estadísticas.** Conjunto de algoritmos y heurísticas utilizados para determinar regresiones de desempeño, perfiles de configuración y toda la información a desplegar en la interfaz de usuario.

Este módulo se implementó en Python y debido a la naturaleza, no es posible proporcionar más detalles, ya que se encuentra bajo un acuerdo de confidencialidad.

- **Interfaz de Usuario Interactiva.** Este módulo incluye todos los elementos necesarios que permiten no solamente visualizar los datos, sino también importar los archivos hacia el repositorio y coordinar la ejecución de los demás módulos del sistema. Se encuentra implementado en Django, con Python y Bootstrap. Las vistas desarrolladas para este módulo siguen principios de UX (experiencia de usuario), manteniendo un diseño intuitivo que permita una fácil interacción con la plataforma, además de cuidar la responsividad de sus elementos.

Esta plataforma se encuentra diseñada para ser utilizada por analistas de desempeño y equipos de desarrollo. Siendo los analistas los encargados de ejecutar diferentes benchmarks y cargar todos los datos generados en la plataforma. Por otro lado, los equipos de desarrollo se encuentran más enfocados en los resultados proporcionados por la herramienta. La información que se despliega incluye reportes, analíticos, tendencias y alertas que pueden ser utilizadas en el proceso de toma de decisiones mientras se encuentran en desarrollo los componentes del sistema en evaluación. Los resultados de desempeño muestran el comportamiento de nuevos componentes/versiones del producto, comparado con versiones anteriores. Al tener acceso a tendencias y analíticos, los desarrolladores pueden evaluar la calidad del producto.

Para validar los beneficios que aporta la plataforma DDP en el ciclo de desarrollo de software, esta ha sido aplicada a un caso de uso, el cual se describe en la siguiente sección.

4. Caso de Uso

DDP ha sido desplegado en un ambiente de desarrollo para cubrir los requerimientos de datos en el proceso de la evaluación de regresión de desempeño de Spectrum Scale. El equipo de desarrollo realiza la evaluación de regresión para validar que la nueva versión mantiene o mejora el desempeño en comparación con versiones previas. La evaluación de regresión de desempeño se compone de múltiples casos de pruebas para datos, metadatos y operaciones, donde benchmarks son usados para estresar el sistema. Estos benchmarks incluyen IOR (Lawrence Livermore National Laboratory, 2018), MDTEST (Lawrence Livermore National Laboratory, 2017), e IOZONE (IOzone, 2018). Cada caso de prueba tiene variaciones y el objetivo principal es crear cargas de trabajo de lectura y escritura para evaluar el desempeño de diferentes componentes de la configuración de Spectrum Scale. Durante la ejecución de los casos de pruebas, múltiples archivos de datos son creados, incluyendo los resultados de los benchmarks, bitácoras de los sistemas de archivos, archivos de configuración y datos de monitoreo de recursos. Después, todos estos datos tienen que ser recolectados, procesados y presentados de forma que muestren tendencias en el desempeño para el equipo de toma de decisiones del área de desarrollo. El ciclo de vida de los datos de este proceso es complejo. Se requiere lidiar con grandes volúmenes de datos, con diferentes formatos y gestionar una considerable cantidad de preprocesamiento para la combinación, agregación y visualización de datos. Para hacer frente a esta complejidad, DDP ha sido desarrollado en tres componentes primarios, como se muestra en la figura 2. Primero, la capa de aplicación aloja todos los submódulos de la interfaz Web que se compone de tableros

interactivos para presentar los resultados al usuario final. Además, la capa de aplicación también aloja todos los algoritmos para los procesos de ETL, combinación y agregación de datos. El componente de almacenamiento de Big Data se integra con una base de datos de Cassandra y con colecciones de MongoDB para agregar todos los datos previamente procesados y almacenarlos de manera no estructurada como una colección de datos de alta flexibilidad. Este desarrollo permite su fácil escalabilidad en cuanto a su capacidad de almacenamiento de acuerdo con los requerimientos de operación. Finalmente, el motor de procesamiento integrado por Spark y Scala se conecta directamente con el componente de almacenamiento de Big Data, procesa los datos y almacena estos nuevos resultados, para que puedan ser utilizados por la interfaz Web. El motor de procesamiento es también altamente escalable para permitir tiempos de procesamiento más eficientes dependiendo del tamaño de las cargas de trabajo.

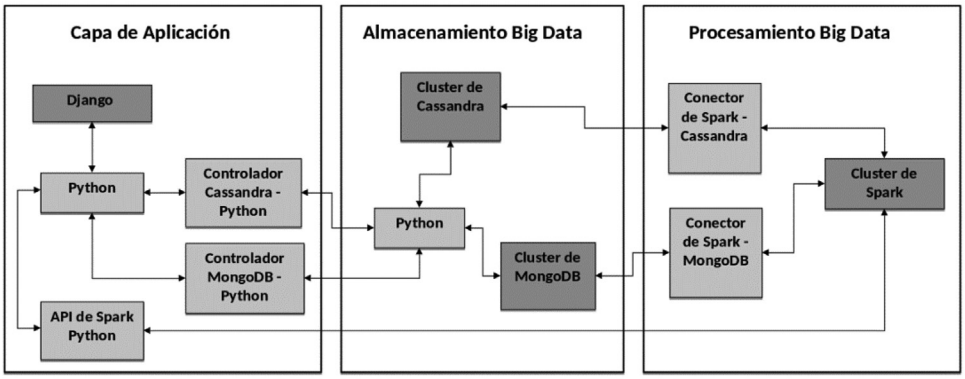


Figura 2 – Estructura general de la plataforma DDP

Esta aplicación mejora de manera significativa el proceso de evaluación de desempeño. Una vez que los datos han sido generados, el analista de desempeño a través de la interfaz interactiva puede, en tan solo unos cuantos clicks, subir los datos. Internamente, el módulo de ETL pre-procesa los datos, los transforma en estructuras apropiadas para su almacenamiento y los carga en el componente de almacenamiento de Big Data. Una vez que los datos han sido almacenados exitosamente, se encuentran disponibles para el motor de procesamiento y para ser explotados. Por lo tanto, en el caso del proceso de evaluación de regresión de desempeño, el analista puede seleccionar las versiones a comparar y obtener los resultados para su evaluación. Esta mejora del proceso permite transparentar toda la complejidad del manejo de datos, reduciendo los tiempos de procesamiento y dejando los datos disponibles para análisis futuros más allá del proceso de evaluación de regresión de desempeño.

Las figuras siguientes muestran algunas interfaces de la aplicación. La figura 3 muestra la ventana donde el analista puede subir los archivos correspondientes a las pruebas de regresión. Estos archivos pueblan la base de datos que se utiliza para generar los reportes de regresión. Todos los datos mostrados en las diferentes interfaces del artículo son ejemplos y no representan resultados reales del proceso de evaluación de desempeño.

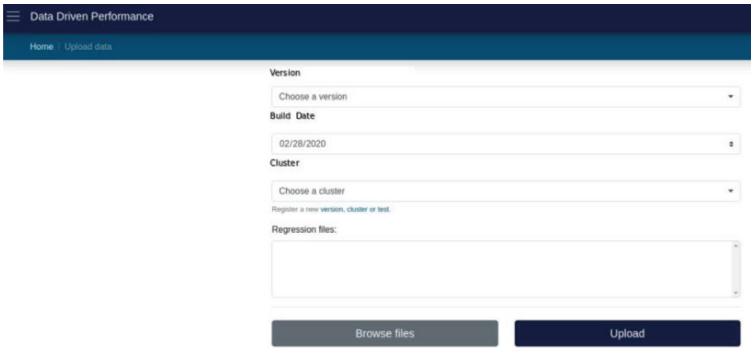


Figura 3 – Interfaz para subir los datos de las pruebas de regresión

En la figura 4 se muestra la interfaz en donde el analista puede seleccionar dos diferentes versiones para comparar y generar un reporte de regresión, seleccionando también el clúster en el cual se corrieron las pruebas, así como la visibilidad o permisos del reporte, es decir, si desea que el reporte sea público o privado.

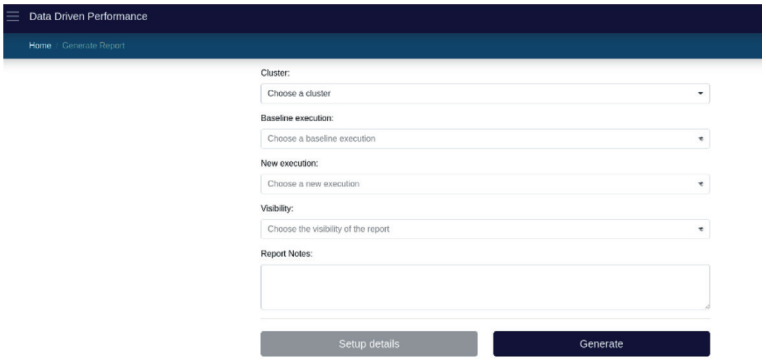


Figura 4 – Interfaz para seleccionar las versiones a comparar para generar el reporte de regresión

En la figura 5 se puede observar un reporte de regresión, aquí el analista puede observar si ha habido alguna mejora o regresión en cuanto al desempeño de la aplicación comparando dos versiones diferentes. Con un click en el botón de “Setup Details”, el analista puede obtener la configuración del clúster que fue utilizado para correr este reporte de regresión.

Con base en este reporte, el equipo de desarrollo puede determinar si existen acciones necesarias que deban ser concretadas antes de liberar la nueva versión del producto, debido a alguna regresión presentada durante las pruebas de desempeño.

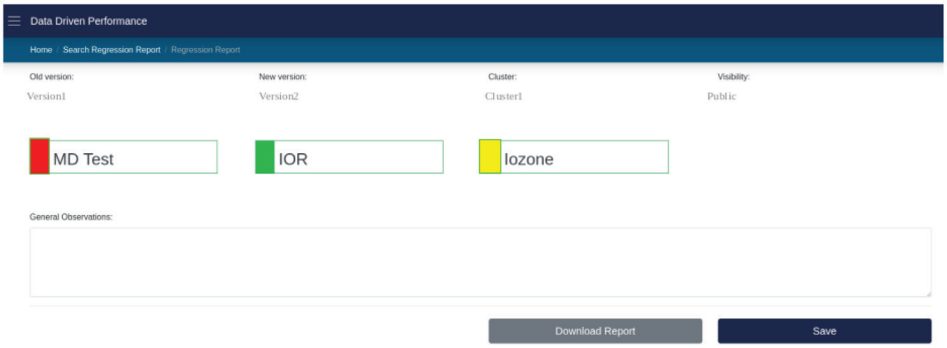


Figura 5 – Reporte de regresión

5. Analíticos de Desempeño en DDP

En cada proceso de desarrollo de software la etapa de pruebas juega un papel primordial. El objetivo de dicha etapa es verificar y validar la calidad del producto que se liberará a los clientes o usuarios finales. Un subconjunto de estas pruebas está relacionado con la evaluación del desempeño del software. El objetivo es detectar problemas que afecten el rendimiento del o los componentes y su interacción con elementos específicos de hardware y aplicaciones de terceros. Los analíticos de desempeño son métricas que permiten observar el rendimiento de componentes de software de forma puntual a través de cifras concretas o elementos visuales.

Algunas de las versiones internas que se generan durante el proceso de desarrollo son elegidas para realizar las pruebas de desempeño. Así tenemos diferentes fechas de pruebas para una misma versión liberada. Cada prueba incluye diferentes benchmarks (MDTest, IOR, IOZONE) y es ejecutada al menos 5 veces. Medimos el número de operaciones por segundo que se pueden realizar, en cada una de las ejecuciones de las pruebas, y tomamos la media para analizar.

Para DDP se eligió realizar dos tipos de análisis de desempeño: un análisis de estadísticos descriptivos y otro de estadísticos predictivos.

En el caso del análisis descriptivo tenemos:

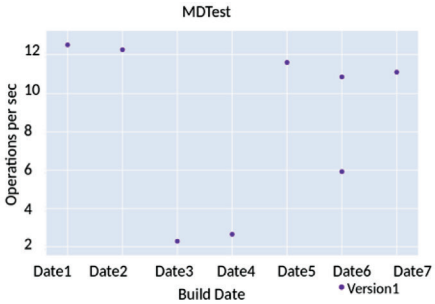
- **Análisis de datos históricos.** Comparamos el desempeño de cada ejecución de las pruebas a través del tiempo.
- **Deltas de desempeño.** Comparación del desempeño de dos versiones internas específicas. Se usa un mapa de calor para representar las deltas del desempeño. El color rojo indica una baja en el desempeño y el color verde una mejora. Se inició esta comparativa únicamente con la versión interna actual y la anterior, pero esto no reflejaba el efecto acumulativo de múltiples decrementos en el desempeño. Así que se decidió calcular esta delta para cualesquiera dos versiones internas, lo que permite obtener una matriz de deltas que es usada para crear el mapa de calor.

- **Resumen de estadísticos descriptivos.** Finalmente tenemos una tabla que nos muestra las tendencias centrales, la dispersión y la forma de la distribución de los datos.

Para el análisis predictivo se eligió un estadístico que mostrará una proyección de las tendencias para predecir los resultados de desempeños en las siguientes versiones.

Las siguientes figuras son algunos ejemplos de los gráficos que se utilizan en el dashboard de análisis de desempeño de la herramienta DDP, esto con la finalidad de encontrar tendencias en los datos para tomar alguna acción necesaria en caso de observar una regresión en el desempeño.

En la figura 6.a (izquierda) se puede observar el desempeño de una operación para un benchmark especifico (MDTest) a lo largo de sus diferentes build dates, con esta gráfica el analista puede detectar alguna tendencia en el desempeño para cada una de las operaciones y dar aviso al equipo de desarrollo. Por su parte, en la figura 6.b (derecha) se observa la gráfica de la distribución del desempeño de una operación específica (Directory creation mean) para comparar el comportamiento de la misma en sus diferentes build dates.



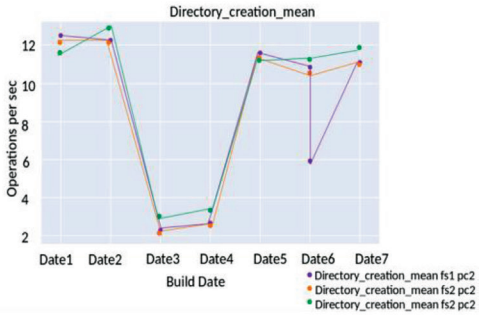
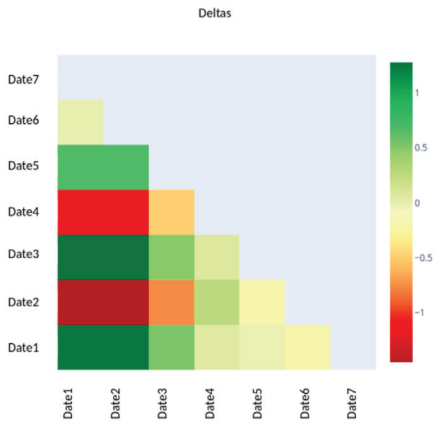


Figura 6.a – Gráfica de dispersión de una versión con respecto a sus bild dates.
Figura 6.b – Gráfica de dispersión de una operación específica, para los diferentes build dates

En la figura 7.a (izquierda) podemos encontrar una comparación entre el desempeño de todas las operaciones de una versión en específico para todos los build dates de la misma, en donde un color verde fuerte indica una mejora y un color rojo intenso indica una regresión. En la Figura7.b (derecha) podemos observar un ejemplo de los estadísticos que se pueden mostrar en el dashboard para ser analizados por el analista de desempeño.

En la actualidad se pueden encontrar muchos ejemplos a nivel mundial sobre los beneficios de analíticos en el proceso de evaluación del desempeño orientado a datos. Un caso interesante es el de Siemens (Siemens Deutschland, 1996 –2020), una fábrica inteligente de la cuarta revolución industrial, donde los productos dirigen su propia fabricación con una producción altamente automatizada, gracias a la analítica de sus datos (euskadi.eus, 2015).



Directory Creation Mean	
Stat	**Value**
Count	8
Mean	6354.32
STD	1120.32
Min	1210.10
25%	4398.16
50%	5439.20
75%	6250.20
Max	7321.30
Variance	1231231.12

Figura 7.a – Gráfica de deltas de una versión específica, para los diferentes build dates.
Figura7.b – Ejemplo de estadísticos usados para el analísis de desempeño

Por otra parte, la plataforma DDP, al contar con un repositorio centralizado que permite almacenar, organizar y visualizar resultados de desempeño, ayuda a los equipos de desarrollo en la toma de decisiones. Esto por medio de reportes, analíticos, tendencias y alertas.

6. Conclusiones y Trabajo Futuro

Las evaluaciones de desempeño de un producto son complejas debido a la variedad en software, hardware, configuración de redes y características específicas de los datos que pueden ser integrados en los sistemas actuales. Un emergente enfoque orientado a datos se presenta en diversas organizaciones que reconocen el valor de los datos como una ventaja competitiva y diferenciador en el mercado, brindando un mayor soporte en la toma de decisiones que mejoren la calidad de sus productos.

Este artículo describe una plataforma basada en un modelo orientado a datos como una propuesta de transformación en el proceso de evaluación de desempeño. Esta plataforma centraliza, almacena, organiza y visualiza los datos de desempeño producidos durante las evaluaciones ejecutadas en el ciclo de desarrollo de un producto. El articulo describe la arquitectura y el conjunto de tecnologías usadas en DDP, así como el proceso de transformación, manipulación y visualización de los datos con la implementación de interfaces interactivas; considerando que estos provienen de diversas fuentes y formatos, por ejemplo: benchmarks, logs, configuraciones de hardware o software. Finalmente, se describe un caso de éxito de la implementación de DDP y las mejoras aportadas a un proceso real de desarrollo de Software.

Algunas de las conclusiones claves que destacamos a partir del desarrollo e implementación de DDP, se listan a continuación:

- Las herramientas orientadas a datos son clave para las mejoras en el proceso de software. Estas pueden ayudar a generar nuevas perspectivas sobre la calidad de

software, adicional a la información que brindan las pruebas funcionales. Una explotación eficiente de los datos puede arrojar tendencias, alertas, modelos predictivos y de desempeño que faciliten la evaluación de la calidad de un producto.

- Una plataforma como DDP acelera el camino a la Inteligencia Artificial. DDP permite recolectar y centralizar los datos de diversos tipos y fuentes. Transformándolos en datos simples y accesibles, organizados en modelos con sentido y fácil de manejar. Los datos se preparan para la fase de análisis, ayudando a los desarrolladores a tomar mejores decisiones que con un impacto positivo en la calidad del producto.
- DDP automatiza la carga de datos y generación de reportes. En el caso de uso se describe como el analista puede visualizar si se presentó una mejora o degradación en el desempeño con solo seleccionar la versión de interés. Hay un ahorro de tiempo implícito que puede ser re-invertido por el analista en la investigación de la causa de los resultados.
- Un modelo orientado a datos con un repositorio central permite explotar los datos históricos e identificar las tendencias entre cualquier periodo de tiempo.
- Un modelo orientado a datos con analíticos y visualizaciones reduce la dependencia de expertos para poder leer y comprender los resultados de las evaluaciones de desempeño.

Actualmente la plataforma cumple el objetivo de automatizar actividades manuales que realizaban los analistas durante las evaluaciones de regresión de desempeño, mejorando así el proceso de desarrollo de software. Se han incorporado los modelos predictivos y sus visualizaciones en dashboards para optimizar el análisis de causa-efecto gracias a los reportes, tendencias, y alertas que despliega la plataforma. Los trabajos futuros se enfocan en profundizar en la explotación de los datos y el desarrollo de modelos predictivos que nos conduzcan a configuraciones del producto que brinden el óptimo desempeño ante diferentes escenarios o modelos cognitivos que permitan anticipar una degradación de desempeño previo a que sea introducido.

Referencias

Aldrich, J. O., & Cunningham, J. B. (2015). *Using IBM® SPSS® Statistics: An Interactive Hands-On Approach*. (S. Publications, Ed.)

Aleti, A., Trubiani, C., van Hoorn, A., & Jamshidi, P. (2018). *An efficient method for uncertainty propagation in robust software performance estimation*. DOI: https://doi.org/10.1016/j.jss.2018.01.010

Dain, J., Forestier, E., Guaitani, P., Guaitani, P., Haas, R., Maestas, C., . . . Sherman, B. (2018). *IBM Software-Defined Storage Guide*. Obtenido de redbooks.ibm.com: http://www.redbooks.ibm.com/abstracts/redp5121.html?Open

Denar, G., Polini, A., & Emmerich, W. (2005). *Performance Testing of Distributed Component Architectures*. DOI: https://doi.org/10.1007/3-540-27071-x_14

Dessau, R., & Bressen Pipper, C. (2007). *"R"-project for statistical computing*. Obtenido de Europe PMC: http://europepmc.org/article/med/18252159#abstract

École Polytechnique Fédérale Lausanne. (2002-2020). *The Scala Programming Language*. Obtenido de scala-lang: https://www.scala-lang.org/

euskadi.eus. (2015). *Visita de Arantxa Tapia a la planta central de SIEMENS en Amberg (Baviera)*. Obtenido de euskadi: https://www.euskadi.eus/gobierno-vasco/contenidos/noticia/2015_10_08_28651/es_28651/28651.html

Foo, K. C., Jiang, Z. M., Adams, B., Hassan, A., Zou, Y., & Flora, P. (2015). An Industrial Case Study on the Automated Detection of Performance Regressions in Heterogeneous Environments. In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. DOI: https://doi.org/10.1109/icse.2015.144

Foo, K. C., Jiang, Z. M., Adams, B., Hassan, E., A., Zou, Y., & & Flora, P. (2015). Automated Detection of Performance Regressions Using Regression Models on Clustered Performance Counters. *In Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, 15–26.

Huijgens, H., Spadini, D., Stevens, D., Visser, N., & van Deursen, A. (2018). Software analytics in continuous delivery: a case study on success factors. In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software*, 1-10.

IBM Hybrid Cloud. (2018). *Accelerate the journey to AI*. Obtenido de ibm: https://www.ibm.com/downloads/cas/MZEA2GKW

IOzone. (2018). *IOzone Filesystem Benchmark*. Obtenido de iozone: http://www.iozone.org

Kim, M., Zimmermann, T., DeLine, R., & Begel, A. (2016). The emerging role of data scientists on software development teams. In*: Proceedings of the 38th International Conference on Software Engineering*, 96–107. DOI: https://doi.org/10.1145/2884781.2884783

Kroß, J., Willnecker, F., Zwickl, T., & Krcmar, H. (2016). PET: continuous performance evaluation tool. In: *Proceedings of the 2nd International Workshop on Quality-Aware DevOps*, 42–43. DOI: https://doi.org/10.1145/2945408.2945418

Lawrence Livermore National Laboratory. (2020). *Used for testing the metadata performance of a file system*. Obtenido de https://github.com/LLNL/mdtest

Lawrence Livermore National Laboratory. (2020). *Parallel filesystem I/O benchmark*. Obtenido de https://github.com/LLNL/ior

Lizy Kurian, J., & Lieven, E. (2018). *Performance Evaluation and Benchmarking*. CRC Press LLC.

Luís Berral, J., Poggi, N., Carrera, D., Call, A., Reinauer, R., & Green, D. (2017). ALOJA: A Framework for Benchmarking and Predictive Analytics in Hadoop Deployments. *IEEE Transactions on Emerging Topics in Computing, 5*, 480-493. DOI: https://doi.org/10.1109/tetc.2015.2496504

Menzies, T., & Zimmermann, T. (2013). Software Analytics: So What? *Software IEEE, 30*, 31-37. Obtenido de ieeexplore.ieee.org: https://ieeexplore.ieee.org/document/6547619

Mockus, A. (2014). *Engineering big data solutions*. DOI: https://doi.org/10.1145/2593882.2593889

mongoDB. (2020). *The database for modern applications*. Obtenido de mongodb: https://www.mongodb.com/

Mostafa, S., Wang, X., & Xie, T. (2017). PerfRanker: prioritization of performance regression tests for collection-intensive software. In: *Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 23–34. DOI: https://doi.org/10.1145/3092703.3092725

Okanović, D., van Hoorn, A., & Zorn, C. (2019). Concern-driven Reporting of Software Performance Analysis Results. In: *Companion of the 2019 ACM/SPEC International Conference on Performance Engineering*, 1-4.

Price, W. (1989). A benchmark tutorial. *IEEE Micro, 9*, 28-43.

Quintero, D., Bolinches, L., Chaudhary, P., Davis, W., Duersch, S., Fachim, C., . . . Weiser, O. (2015). *IBM Spectrum Scale (formerly GPFS)*. Obtenido de http://www.redbooks.ibm.com/redbooks/pdfs/sg248254.pdf

Anbunathan, R., & Basu, A. (2015). Data driven architecture based automated test generation for Android mobile. *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. DOI: https://doi.org/10.1109/iccic.2015.7435772

Siemens Deutschland. (1996 – 2020). *Unser Anliegen: eine bessere Zukunft für alle*. Obtenido de new.siemens: https://new.siemens.com/de/de/unternehmen/jobs.html

The Apache Software Foundation. (2016). *Manage massive amounts of data, fast, without losing sleep*. Obtenido de cassandra.apache: http://cassandra.apache.org/

The Apache Software Foundation. (2018). *Apache Spark™ is a unified analytics engine for large-scale data processing*. Obtenido de spark.apache: https://spark.apache.org

Velcu-Laitinen, O., & Yigitbasioglu, O. (2012). The Use of Dashboards in Performance Management: Evidence from Sales Managers. *The International Journal of Digital Accounting Research, 12*.

Yandrapally, R., Thummalapenta, S., Sinha, S., & Chandra, S. (2014). Robust test automation using contextual clues. In: *Proceedings of the 2014 International Symposium on Software Testing and Analysis*, 304. DOI: https://doi.org/10.1145/2610384.2610390

Yigitbasioglu, O., & Velcu, O. (2012). *A review of dashboards in performance management: Implications for design and research*. DOI: https://doi.org/10.1016/j.accinf.2011.08.002